

Refine Search

Search Results -

Terms	Documents
L6 and (virus near checker)	5

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

L21

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Saturday, January 08, 2005 [Printable Copy](#) [Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u>
side by side			result set
DB=USPT; PLUR=NO; OP=OR			
<u>L21</u>	L6 and (virus near checker)	5	<u>L21</u>
<u>L20</u>	L6 and (virus near scan)	35	<u>L20</u>
<u>L19</u>	L17 AND update	37	<u>L19</u>
<u>L18</u>	L17 and 713/\$\$\$.ccls.	6	<u>L18</u>
<u>L17</u>	L15 and (rollback.ab. or undo.ab. or previous.ab. or previous.ab. or version.ab.)	48	<u>L17</u>
<u>L16</u>	L15 and rollback.ab. or undo.ab. or previous.ab. or previous.ab. or version.ab.	16042	<u>L16</u>
<u>L15</u>	L14 AND (download OR deploy or transfer)	118	<u>L15</u>
<u>L14</u>	L13 and (rollback OR ((old or prior or previous) ADJ (version)))	195	<u>L14</u>
<u>L13</u>	l3 or l6	1503	<u>L13</u>
<u>L12</u>	L6 and beta and virus and detect	1	<u>L12</u>
<u>L11</u>	L9 AND previous	9	<u>L11</u>

<u>L10</u>	L9 and rollback	1	<u>L10</u>
<u>L9</u>	L8 AND update	18	<u>L9</u>
<u>L8</u>	L7 AND (virus ADJ scan)	30	<u>L8</u>
<u>L7</u>	L6 AND virus	150	<u>L7</u>
<u>L6</u>	713/\$\$\$.ccls. AND scan	1334	<u>L6</u>
<u>L5</u>	L4 and install	51	<u>L5</u>
<u>L4</u>	L3 AND (download OR deploy or transfer)	99	<u>L4</u>
<u>L3</u>	L1 and (rollback OR ((old or prior or previous) ADJ (version)))	170	<u>L3</u>
<u>L2</u>	L1 and rollback OR ((old or prior or previous) ADJ (version))	1752	<u>L2</u>
<u>L1</u>	717/168-178.cccls.	948	<u>L1</u>

END OF SEARCH HISTORY

Hit List

[Clear](#)[Generate Collection](#)[Print](#)[Fwd Refs](#)[Bkwd Refs](#)[Generate OACS](#)

Search Results - Record(s) 1 through 50 of 51 returned.

1. Document ID: US 6836886 B2

L5: Entry 1 of 51

File: USPT

Dec 28, 2004

US-PAT-NO: 6836886

DOCUMENT-IDENTIFIER: US 6836886 B2

TITLE: Method and apparatus for delivering electronic information

DATE-ISSUED: December 28, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tuerke; Thomas Mike	Novato	CA		
Rawat; Priya	Richmond	CA		
Huang; Janice Pei-I	San Jose	CA		

US-CL-CURRENT: 717/178; 707/10, 707/100, 707/4, 715/501.1, 715/513, 717/105,
717/115

ABSTRACT:

A method, apparatus, system, and article of manufacture for electronically obtaining a Web page in a Web browser. A Web page is requested. In response, the browser receives a bootstrap file that comprises a declaration of a component module control object, and a script block comprising a method that accesses the component module control object. The object identified in the declaration is obtained. The object also includes one or more resources. The object is then installed in the Web browser. The Web browser may then use the method of the script block to extract a Web page based on the resources of the component module control object. The Web browser may also be configured to determine if the control object is an up to date version. If the control object is not an up to date version, the browser obtains and installs an up to date version.

27 Claims, 2 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 2

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Detailed Description](#) | [Claims](#) | [KDDC](#) | [Drawn D](#)

2. Document ID: US 6836794 B1

L5: Entry 2 of 51

File: USPT

Dec 28, 2004

US-PAT-NO: 6836794
DOCUMENT-IDENTIFIER: US 6836794 B1

TITLE: Method and system for assigning and publishing applications

DATE-ISSUED: December 28, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lucovsky; Mark	Redmond	WA		
Cherry; Michael J.	Redmond	WA		
Plastina; Daniel	Issaquah	WA		
Shah; Bharat	Bellevue	WA		
Mishra; Debi P.	Redmond	WA		
Kays, Jr.; David E.	Carnation	WA		
Horstmann; Markus	Redmond	WA		

US-CL-CURRENT: 709/223; 717/177

ABSTRACT:

A method and system for managing and deploying applications across a computer network by assigning and publishing applications to user and computer policy recipients. Assigned applications are automatically applied via a script to the policy recipients, thereby ensuring that the recipient has an administrator-specified workstation configuration. Assigned applications are advertised so as to appear available to the user, by adding an application shortcut to the start menu, and by populating the machine registry with appropriate application information. Other applications may be published to users, whereby those applications are optionally available for use by users. Published application information is maintained in a centralized store of information on the network. Assigned and published applications may be installed on demand, such as when the application is activated.

41 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Specification](#) | [Claims](#) | [KOMC](#) | [Draw. D.](#)

3. Document ID: US 6832373 B2

L5: Entry 3 of 51

File: USPT

Dec 14, 2004

US-PAT-NO: 6832373
DOCUMENT-IDENTIFIER: US 6832373 B2

TITLE: System and method for updating and distributing information

DATE-ISSUED: December 14, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Neill; Patrick J.	Dana Point	CA		

US-CL-CURRENT: 717/171; 714/25, 714/807

ABSTRACT:

The present invention discloses efficient devices, systems, and methods for updating digital information sequences that are comprised by software (110a), devices (104c), and data (110c). In addition, these digital information sequences may be stored and used in various forms, including but not limited to files, memory locations, and/or embedded storage locations. The disclosed invention is thus suitable for updating many types of digital information sequences and in the context of updating software comprised of multiple files. Furthermore, the devices, systems, and methods described herein provide a developer skilled in the art with an improved ability to generate update information as needed and, additionally, allow users to proceed through a simplified update path, which is not error-prone, and may be performed more quickly than through the use of existing technologies.

55 Claims, 20 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 16

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KINDC](#) | [Drawn D.](#)

4. Document ID: US 6804663 B1

L5: Entry 4 of 51

File: USPT

Oct 12, 2004

US-PAT-NO: 6804663

DOCUMENT-IDENTIFIER: US 6804663 B1

TITLE: Methods for optimizing the installation of a software product onto a target computer system

DATE-ISSUED: October 12, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Delo; John C.	Bellevue	WA		

US-CL-CURRENT: 707/3; 707/10, 707/100, 707/101, 707/201, 709/226, 709/229, 717/174

ABSTRACT:

Database tables may be created in a modular fashion and maybe efficiently merged together. Differences between various versions of software products also be recorded in database transforms. The database transforms can then be applied to a prior installation database, so as to upgrade the prior installation database. In this manner, enhancements, patches, upgrades, custom installations, can be applied in modular fashion without the need to ship an entire modified installation database. Database transforms created from a first version and an upgraded first version can also be used to upgrade corresponding second versions into an upgraded

second versions.

10 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Claims](#) | [KMC](#) | [Drawn D.](#)

5. Document ID: US 6754896 B2

L5: Entry 5 of 51

File: USPT

Jun 22, 2004

US-PAT-NO: 6754896
DOCUMENT-IDENTIFIER: US 6754896 B2

TITLE: Method and system for on-demand installation of software implementations

DATE-ISSUED: June 22, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mishra; Debi P.	Redmond	WA		
Kays, Jr.; David E.	Carnation	WA		
Horstmann; Markus	Redmond	WA		
Lucovsky; Mark H.	Redmond	WA		
Chan; Shannon J.	Bellevue	WA		
Shah; Bharat A.	Bellevue	WA		
Jensenworth; Gregory A.	Redmond	WA		

US-CL-CURRENT: 717/176; 709/224

ABSTRACT:

A method and system for installing software implementations such as applications and COM classes as they are needed from an external source, such as a centralized network store. When a software implementation is needed, the system and method first look to the local system (e.g., registry) for that software implementation, and if found, returns the information such as a local path needed to use the software implementation. If the implementation is not found locally, the present invention dynamically looks to a centralized class store of a network, to locate the needed implementation. When located, the implementation is downloaded and locally installed in a manner that is essentially transparent to the user. Software implementations such as application products may be divided into features and components to improve on-demand installation thereof.

34 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Claims](#) | [KMC](#) | [Drawn D.](#)

6. Document ID: US 6751658 B1

L5: Entry 6 of 51

File: USPT

Jun 15, 2004

US-PAT-NO: 6751658

DOCUMENT-IDENTIFIER: US 6751658 B1

TITLE: Providing a reliable operating system for clients of a net-booted environment

DATE-ISSUED: June 15, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Haun; Christopher K.	San Jose	CA		
Prouse; Craig Harvey	Mountain View	CA		
Sokol, Jr.; Joseph	San Jose	CA		
Resch; Paul M.	Santa Clara	CA		

US-CL-CURRENT: 709/222; 709/203, 709/208, 709/220, 709/227, 713/1, 713/2, 717/172

ABSTRACT:

A method and apparatus are provided for supplying a reliable and maintainable operating system in a net-booted environment. According to one embodiment, a network computer (NC) client boots from a boot image provided by an NC server. The boot image includes information identifying the location of one or more system volumes on the NC server that contain operating system software. In response to an attempt to modify the contents of the one or more system volumes, the NC client causes information identifying the modification to be recorded on the NC server separate from the one or more system volumes in a storage area associated with the NC client.

11 Claims, 12 Drawing figures

Exemplary Claim Number: 8

Number of Drawing Sheets: 11

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KOMC](#) | [Drawn D.](#)

7. Document ID: US 6735766 B1

L5: Entry 7 of 51

File: USPT

May 11, 2004

US-PAT-NO: 6735766

DOCUMENT-IDENTIFIER: US 6735766 B1

**** See image for Certificate of Correction ****

TITLE: Method and computer-readable medium for installing an upgrade to an application program

DATE-ISSUED: May 11, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chamberlain; Benjamin C.	Redmond	WA		
Collie; Robert C.	Redmond	WA		
Hatakeda; Darrin N.	Redmond	WA		

US-CL-CURRENT: 717/173; 709/203, 709/218, 709/219, 717/177

ABSTRACT:

A method and computer-readable medium for upgrading an application program, and making an installer program aware that an application has been upgraded is disclosed. The method recognizes that an upgrade to the application has been initiated. The upgrade includes an identifier for the application to be upgraded, as well as information required for the installer to perform the upgrade. The installer program accesses the upgrade information, which includes instructions necessary for installing the upgrade. The installer program also accesses information related to the installed state of the application and related applications. The installer program determines whether or not the upgrade should be performed. If the upgrade should be performed, the installer program performs the upgrade and stores the fact that the application has been upgraded. If the application should not be upgraded, the installer program stores information reflecting the fact that the upgrade is available but not installed.

7 Claims, 10 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 10

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KWMC](#) | [Drawn D](#)

 8. Document ID: US 6698018 B1

L5: Entry 8 of 51

File: USPT

Feb 24, 2004

US-PAT-NO: 6698018

DOCUMENT-IDENTIFIER: US 6698018 B1

TITLE: System and method of multiple-stage installation of a suite of applications

DATE-ISSUED: February 24, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Zimniewicz; Jeff A.	Bellevue	WA		
Helgeson; Ryan	Kirkland	WA		
Marino; Phillip J.	Dublin	OH		
Johnson; Crista E.	Seattle	WA		

US-CL-CURRENT: 717/175; 717/169, 717/171, 717/176

ABSTRACT:

A multiple stage installation system for the installation and setup of a suite of applications segregates and organizes the preparation, installation, clean up, optimization, etc. into functional groupings that define the multiple stages of the installation process. These functional groupings include actions to be performed on behalf of and to any and all of the applications to be installed. While different stages may be defined, the system preferably includes a pre-install phase, an install phase, and a post-install phase during which different functional activities are performed. An optimization phase may also be included to allow optimization of applications that have already been installed. In a preferred embodiment, the applications to be installed implement a COM interface that contains a method for each stage supported. The core installation system determines the installation order for the applications, acquires the COM interface from the application, and for each install stage, calls the appropriate method on that interface. Each application's method for a given stage is called before any methods of any applications for the next stage.

14 Claims, 8 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

Full Title Citation Front Review Classification Date Reference Subject Description Claims KCNC Drawn On

9. Document ID: US 6698017 B1

L5: Entry 9 of 51

File: USPT

Feb 24, 2004

US-PAT-NO: 6698017

DOCUMENT-IDENTIFIER: US 6698017 B1

TITLE: Software migration on an active processing element

DATE-ISSUED: February 24, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Adamovits; Peter Joseph	Carleton Place			CA
Dysart; Keith Clifford	Stittsville			CA
Berube; Louis Pierre	Carp			CA

US-CL-CURRENT: 717/168

ABSTRACT:

A method, apparatus, and system are provided for performing a migration of control over a processing element from an original software system to a replacement software system. A replacement software system in memory associated with an active processing element is configured while an original software system controls the active processing element. The replacement software system is prepared to take control of the active processing element when state information is communicated to it from the original software system. Control of the active processing element is transferred to the replacement software system by a migration manager. A virtual machine may be used to facilitate the transfer of control, by interfacing between the active processing element, the original software system and the replacement software system. The virtual machine and original software system may be

deactivated once control is transferred.

54 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Assignee](#) | [Inventor](#) | [Claims](#) | [KMC](#) | [Drawn D](#)

10. Document ID: US 6681390 B2

L5: Entry 10 of 51

File: USPT

Jan 20, 2004

US-PAT-NO: 6681390
DOCUMENT-IDENTIFIER: US 6681390 B2
** See image for Certificate of Correction **

TITLE: Upgrade of a program

DATE-ISSUED: January 20, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fiske; Robert	Wayland	MA		

US-CL-CURRENT: 717/173; 713/1, 713/2, 714/10, 714/13, 717/169, 717/170, 717/171,
717/172, 717/175, 717/176

ABSTRACT:

A method and processor program product for performing an upgrade of a program on a processor are provided. An upgraded version of the program is received into the processor and a backup of the program is created in memory associated with the processor. The upgraded version of the program is then installed and the processor is rebooted. Prior to initializing other application drivers, a reboot driver checks if the state of the processor is correct. If the reboot driver determines that the state of the processor is incorrect, the system reverts to the backup of the program. In alternative embodiments, if the reboot driver determines that the state is incorrect, the system inhibits the loading of other application drivers. Additionally, the reboot driver may set a counter to indicate the number of times that the processor is rebooted. If the counter has not reached a predetermined threshold, the system attempts to reboot the processor with the upgraded version of the program. If the counter reaches the predetermined threshold, the system reverts to the backup of the program.

56 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Assignee](#) | [Inventor](#) | [Claims](#) | [KMC](#) | [Drawn D](#)

11. Document ID: US 6675382 B1

US-PAT-NO: 6675382
DOCUMENT-IDENTIFIER: US 6675382 B1

TITLE: Software packaging and distribution system

DATE-ISSUED: January 6, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Foster; Gary D.	Santa Clara	CA		

US-CL-CURRENT: 717/177; 707/1, 707/10, 717/169, 717/170, 717/172, 717/175

ABSTRACT:

A method and apparatus for packaging and distributing software. Embodiments of the invention comprise a software packaging system that is portable across many platforms. Each package is self-contained in form of a single-file entity that comprises a payload file and a control file. The payload file is an archive file that contains a compressed collection of all the software files that are required for installation of the software package. The control file includes the necessary information for installation of the files contained in the payload file, in addition to other descriptive information used to determine the size, type, location of storage, and other useful attributes of a software package, even before it is installed on a system. Security measures have been implemented in the system to detect a package the contents of which have been tampered with. Embodiments of the invention can be utilized to install packaged software that is accessible via the Internet. A package on a remote source can be accessed and installed using a Uniform Resource Locator (URL) that indicates the package's specific address on the remote source. Embodiments of the invention are designed such that the entire system is small in size so that the storage space and the transmission bandwidth required for their storage or transportation are minimized. Embodiments of the invention may be used to install, remove or update a software package.

5 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

Full	Title	Citation	Front	Review	Classification	Date	Reference	Examiner	Attorney	Claims	KMC	Draw
------	-------	----------	-------	--------	----------------	------	-----------	----------	----------	--------	-----	------

12. Document ID: US 6651249 B2

US-PAT-NO: 6651249
DOCUMENT-IDENTIFIER: US 6651249 B2

TITLE: Multi-tiered incremental software updating

DATE-ISSUED: November 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Waldin; Ray Soon	Los Angeles	CA		
Nachenberg; Carey	Northridge	CA		

US-CL-CURRENT: 717/170; 717/168, 717/169, 717/171, 717/172**ABSTRACT:**

A software application (110) is updated to a newer version by means of incremental update patches (122). The incremental update patches (122) each contain that information necessary to transform one version of an application to another version. Any version of an application (110) may be upgraded to any other version of the application, through the use of a series of incremental update patches (122). The appropriate incremental update patches (122) are distributed in a multi-tiered manner, such that some update patches (122) update the application (110) by only one version, and others update the application (110) by several versions.

3 Claims, 9 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draft
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	-------

 13. Document ID: US 6604236 B1

L5: Entry 13 of 51

File: USPT

Aug 5, 2003

US-PAT-NO: 6604236

DOCUMENT-IDENTIFIER: US 6604236 B1

TITLE: System and method for generating file updates for files stored on read-only media

DATE-ISSUED: August 5, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Draper; Stephen Peter Willis	Hants			GB
Collins; Brian James	Surrey			GB
Falls; Patrick Terence	Berkshire			GB

US-CL-CURRENT: 717/170**ABSTRACT:**

A method and system of the present invention generates a representation of a new version of an original file system with reference to the original file system and the new version of the file system. Use of data from previous versions of the file system reduces the amount of data to be stored in the delta directory map file, delta modification data block file, and delta look up table generated for the data portions unique to the newest version of an original file system. The inventive

process produces delta data block records that identify the location of data portions that may be used to generate the newest version of the file system. The data portions may be located in a file in the original file system, a delta modification data block file in a previous version of the original file system or a delta modification data block file for the newest version of the original file system.

The method of the present invention is performed by generating a basis index table identifying the data content of an original file system, generating a file of modification data blocks that may be used to modify the data content of the original file system, and generating a delta look up table that identifies the location of the data blocks used to represent the newest version of the original file system. The delta look up table and the file of modification data blocks may be stored for delivery to a computer on which a copy of the original file system is stored. The delta look up table and the file of modification data blocks are then used by the computer system on which a copy of the original file system is stored to provide the data content for a new version of the original file system. This is done in way that appears to provide a single file system containing the new version of the file system. Thus, the method of the present invention may be used to generate data for updating the content of a copy of the original file system without having to generate a copy of every file and data block for the new content of the original file system.

164 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KWMC](#) | [Draw. D.](#)

14. Document ID: US 6550062 B2

L5: Entry 14 of 51

File: USPT

Apr 15, 2003

US-PAT-NO: 6550062

DOCUMENT-IDENTIFIER: US 6550062 B2

TITLE: System and method for launching generic download processing in a computer build-to-order environment

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Barajas; Gaston M.	Cedar Park	TX		
Nix; Todd	Austin	TX		

US-CL-CURRENT: 717/178

ABSTRACT:

A system and method for installing and/or testing software for a build-to-order computer system having a plurality of components includes a plurality of elements. The computer system has an associated step sequence. The step sequence includes a plurality of steps where each step includes at least one command and a step is

associated with a respective software component descriptor. The at least one command can be in more than one computer software language. A software component descriptor corresponds to installing a respective software component on the computer system. The method includes accessing the at least one command within the step sequence; determining a corresponding computer software language associated with the at least one command; executing the at least one command in the corresponding computer software language to install the respective software component; and repeating the accessing, determining and executing for the plurality of steps.

30 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KMC](#) | [Drawn D.](#)

15. Document ID: US 6523166 B1

L5: Entry 15 of 51

File: USPT

Feb 18, 2003

US-PAT-NO: 6523166

DOCUMENT-IDENTIFIER: US 6523166 B1

TITLE: Method and system for on-demand installation of software implementations

DATE-ISSUED: February 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mishra; Debi P.	Redmond	WA		
Kays, Jr.; David E.	Carnation	WA		
Horstmann; Markus	Redmond	WA		
Lucovsky; Mark H.	Redmond	WA		
Chan; Shannon J.	Bellevue	WA		
Shah; Bharat A.	Bellevue	WA		
Jensenworth; Gregory A.	Redmond	WA		

US-CL-CURRENT: 717/174; 707/104.1, 709/203, 709/246

ABSTRACT:

A method and system for installing software implementations such as applications and COM classes as they are needed from an external source, such as a centralized network store. When a software implementation is needed, the system and method first look to the local system (e.g., registry) for that software implementation, and if found, returns the information such as a local path needed to use the software implementation. If the implementation is not found locally, the present invention dynamically looks to a centralized class store of a network, to locate the needed implementation. When located, the implementation is downloaded and locally installed in a manner that is essentially transparent to the user. Software implementations such as application products may be divided into features and components to improve on-demand installation thereof.

44 Claims, 10 Drawing figures

Exemplary Claim Number: 41

Number of Drawing Sheets: 9

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Description](#) | [Claims](#) | [KWMC](#) | [Drawn On](#)

16. Document ID: US 6493871 B1

L5: Entry 16 of 51

File: USPT

Dec 10, 2002

US-PAT-NO: 6493871

DOCUMENT-IDENTIFIER: US 6493871 B1

TITLE: Method and system for downloading updates for software installation

DATE-ISSUED: December 10, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McGuire; Thomas D.	Woodinville	WA		
Sliger; Michael V.	Issaquah	WA		
Welch; Daniel C.	Redmond	WA		
Vishnumurty; Rajendra H.	Bellevue	WA		
Aul; Gabriel J.	Seattle	WA		
Wallace; Oliver I.	Redmond	WA		
Nichols; Gregory W.	Seattle	WA		
Auerbach; Alan B.	Redmond	WA		

US-CL-CURRENT: 717/173; 717/169, 717/171, 717/172, 717/175, 717/178

ABSTRACT:

A method and system for downloading software update data for installing a revised software product on a client computer minimizes the amount of update data to be transmitted over the network by downloading only those files needed to put the client computer in the state for installing the product. In the beginning of the downloading process, the client computer obtains from a setup server an initial setup package that includes a setup program and a list of files required for installing the software product. The setup program running on the client computer then determines whether some current or earlier versions of those files required for installation already exist on the client computer, and compiles a request list of files needed for updating the client computer. The client computer sends the request list to a download server, which maintains a collection of update files and patches. In response to the request list, the download server downloads updating files to the client. Depending of the availability of the requested files or other factors, the downloaded files may or may not be exactly those requested. Using the downloaded files, the setup program updates the existing files to provide the set of installation files on the client computer. The desired revised software product is then installed on the client computer.

50 Claims, 12 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 12

17. Document ID: US 6442754 B1

L5: Entry 17 of 51

File: USPT

Aug 27, 2002

US-PAT-NO: 6442754

DOCUMENT-IDENTIFIER: US 6442754 B1

TITLE: System, method, and program for checking dependencies of installed software components during installation or uninstallation of software

DATE-ISSUED: August 27, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Curtis; Bryce Allen	Round Rock	TX		

US-CL-CURRENT: 717/175; 707/104.1

ABSTRACT:

Disclosed is a method, system, program, and data structure for installing a program onto a computer including an operating system. Dependency objects indicate a dependent component on which the program to install depends. The program processes the dependency objects before installing the program and determines an operating system command that is capable of determining whether the dependent component indicated in the dependency object is installed in the computer. The program then executes the operating system command to determine whether the dependent components indicated in the dependency objects are installed in the computer. An indication is made as to the dependent components that are not installed after determining that dependent components are not installed.

31 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

18. Document ID: US 6438749 B1

L5: Entry 18 of 51

File: USPT

Aug 20, 2002

US-PAT-NO: 6438749

DOCUMENT-IDENTIFIER: US 6438749 B1

** See image for Certificate of Correction **

TITLE: Method and system for restoring a computer to its original state after an unsuccessful patch installation attempt

DATE-ISSUED: August 20, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chamberlain; Benjamin C.	Redmond	WA		

US-CL-CURRENT: 717/174; 714/15, 714/2

ABSTRACT:

A system, method and computer-readable medium for creating patch rollback scripts for operations performed by an installer program module. A patch rollback script may be used to reverse an action of an installer program module so that a user's machine is restored to its original state, i.e., the state before the installer program module began performing patch operations. In performing patch operations, the installer program module typically generates patch installation script records of specific actions that are to be performed to a particular machine with regard to patch installation data. During the processing of each patch installation script record, an inverse record is created, known as a patch rollback script record. When installing or modifying a file, if there is no file with the same name in the same location stored on the machine, a patch rollback script record describing how to delete the file is stored. However, if an old file stored in the same location does exist, the old file is backed up to another location and a patch rollback script record is saved describing how to copy the old file back to its prior location on the machine.

16 Claims, 10 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 9

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWOC	Drawn
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	-------

19. Document ID: US 6418554 B1

L5: Entry 19 of 51

File: USPT

Jul 9, 2002

US-PAT-NO: 6418554

DOCUMENT-IDENTIFIER: US 6418554 B1

**** See image for Certificate of Correction ****

TITLE: Software implementation installer mechanism

DATE-ISSUED: July 9, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Delo; John C.	Bellevue	WA		
Haar; Malcolm S.	Seattle	WA		
Parulekar; Chetan A.	Redmond	WA		
Ferrier; Tracy D.	Issaquah	WA		
Chamberlain; Benjamin	Redmond	WA		
Gonzalez; David E.	Issaquah	WA		

McKinnis; David R.

Seattle WA

US-CL-CURRENT: 717/174

ABSTRACT:

A method and mechanism for automatically installing software implementations such as applications and COM classes as they are needed from an external source. When a software implementation is needed, the mechanism first looks to the local system (e.g., registry) for that software implementation, and if found, returns the information such as a local path needed to use the software implementation. If the implementation is not found, the mechanism looks to another source, such as a CD-ROM or a centralized class store of a network, to locate the needed implementation. When located, the implementation is downloaded and locally installed from the source, and a local path is returned in a manner that is essentially transparent to the user. Software implementations such as application products may be divided into features and components to improve on-demand installation thereof.

37 Claims, 16 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 15

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KMC](#) | [Draw. D.](#)

20. Document ID: US 6408434 B1

L5: Entry 20 of 51

File: USPT

Jun 18, 2002

US-PAT-NO: 6408434

DOCUMENT-IDENTIFIER: US 6408434 B1

TITLE: System and method for using a substitute directory to automatically install an update program

DATE-ISSUED: June 18, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fujiwara; Nobuyuki	Kanagawa			JP

US-CL-CURRENT: 717/170; 717/175, 717/176

ABSTRACT:

A system and method for using a substitute directory to automatically install an update program preferably comprises a plurality of computer systems that are connected to a distributed network such as the Internet. Download modules on the computer systems automatically access and download update programs from the distributed network. Install modules then automatically create a unique substitute directory and install the downloaded update programs into the substitute directory to complete the software installation procedure, in accordance with the present invention.

38 Claims, 10 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 10

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KWMC](#) | [Drawn D.](#)

21. Document ID: US 6378128 B1

L5: Entry 21 of 51

File: USPT

Apr 23, 2002

US-PAT-NO: 6378128

DOCUMENT-IDENTIFIER: US 6378128 B1

** See image for Certificate of Correction **

TITLE: System and method for dynamically modifying an install-set

DATE-ISSUED: April 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edelstein; Noah B.	Seattle	WA		
Kanerva; Heikki J.	Redmond	WA		

US-CL-CURRENT: 717/174; 709/221

ABSTRACT:

Dynamically creating or modifying an install-set of a program module comprising required components and optional components. Usage and configuration parameters are detected in the operating environment of the target computer system. Usage parameters relate to past usage of prior versions of the components of the program module. Configuration parameters may be any configuration aspect of the operating environment, including hardware and software configuration. Based on detected usage and/or configuration parameters, determinations may be made as to whether each of the components of the program module are to be included in, or excluded from, an install-set. The install-set is the set of required and/or optional components that are selected for installation onto the target computer system. Based on the determinations, the install preferences of each component are set. Install preferences are properties that signal whether each component is to be installed by an installer module. The installer module installs, or does not install, the components according to their install preference.

32 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KWMC](#) | [Drawn D.](#)

22. Document ID: US 6378127 B1

L5: Entry 22 of 51

File: USPT

Apr 23, 2002

US-PAT-NO: 6378127
DOCUMENT-IDENTIFIER: US 6378127 B1
**** See image for Certificate of Correction ****

TITLE: Software installation and validation using custom actions

DATE-ISSUED: April 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Delo; John C.	Bellevue	WA		

US-CL-CURRENT: 717/174

ABSTRACT:

A software installation and semantic database validation system using custom actions is disclosed. The system includes a database engine module for maintaining a database. The database contains a custom action table, with columns in the table providing information about the action. Each row in the table represent an individual action. The system also includes an installation engine module operative to read an action value from an action column of the action row and causing an action specified by the action value to be performed by a computer. The actions can be executable programs, Dynamic Link Library modules, or script written in scripting languages such as JavaScript or Visual Basic.

2 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Claims](#) | [KWMC](#) | [Drawn](#) | [D](#)

23. Document ID: US 6377960 B1

L5: Entry 23 of 51

File: USPT

Apr 23, 2002

US-PAT-NO: 6377960

DOCUMENT-IDENTIFIER: US 6377960 B1

TITLE: Transactional configuration store and runtime versus administration isolation with version snapshots and aging

DATE-ISSUED: April 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Qiu; Wenjun	Bellevue	WA		
Zander; Jason L.	Redmond	WA		
Horstmann; Markus	Redmond	WA		
Devlin; William D.	Redmond	WA		

US-CL-CURRENT: 707/203; 707/200, 717/170, 717/174

ABSTRACT:

An improved registration datastore comprises a datastore containing the database coupled to a data table object structure to present the data to a registration system in the form of an abstract table of data. The use of a data table structure between the registration system and the datastore provides storage location and format independence as the data table object presents the registration data to a calling object in the form of a data level table, a collection of configuration data items. The improved registration system permits one or more objects to be simultaneously installed into the registration database by different sources. The new registration system utilizes a database versioning and aging mechanism to permit multiple calling objects to operate using a version of the database known to be valid when its operations began.

15 Claims, 9 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 8

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KWMC](#) | [Drawn Ds](#)

24. Document ID: US 6363499 B1

L5: Entry 24 of 51

File: USPT

Mar 26, 2002

US-PAT-NO: 6363499

DOCUMENT-IDENTIFIER: US 6363499 B1

**** See image for Certificate of Correction ****

TITLE: Method and system for restoring a computer to its original state after an unsuccessful installation attempt

DATE-ISSUED: March 26, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Delo; John C.	Bellevue	WA		
Chamberlain; Benjamin	Redmond	WA		

US-CL-CURRENT: 714/15; 717/126, 717/174

ABSTRACT:

A system and method for creating rollback scripts for operations performed by an installer program module. A rollback script may be used to reverse an action of an installer program module so that a user's machine is restored to its original state, i.e., the state before the installer program module began performing operations. In performing operations, the installer program module typically generates installation script records of specific actions that are to be performed to a particular machine with regard to installation data. During the processing of each installation script record, an inverse record is created, known as a rollback script record. When installing a file, if there is no file with the same name in the same location stored on the machine, then a rollback script record describing how to delete the file is stored. However, if an old file stored in the same location does exist, then the old file is backed up to another location and a

rollback script record is saved describing how to copy the old file back to its prior location on the machine.

15 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KMC](#) | [Drawn D...](#)

25. Document ID: US 6332217 B1

L5: Entry 25 of 51

File: USPT

Dec 18, 2001

US-PAT-NO: 6332217

DOCUMENT-IDENTIFIER: US 6332217 B1

TITLE: Software inventory control system

DATE-ISSUED: December 18, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hastings; Lawrence C.	Mountain View	CA		

US-CL-CURRENT: 717/178; 707/1

ABSTRACT:

A method for addressing problems of computer remote software configuration management is provided wherein a computer program executes a scripting language in order to examine the software installed on a computer. In the event the script determines that software needs to be downloaded, installed or configured, it causes programs to be executed to perform these functions. Such a program enables a user to consistently get correct software installed with the correct configuration for the system, thus reducing demands for technical support and the costs associated therewith.

6 Claims, 1 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 1

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Advanced Search](#) | [Claims](#) | [KMC](#) | [Drawn D...](#)

26. Document ID: US 6324692 B1

L5: Entry 26 of 51

File: USPT

Nov 27, 2001

US-PAT-NO: 6324692

DOCUMENT-IDENTIFIER: US 6324692 B1

TITLE: Upgrade of a program

DATE-ISSUED: November 27, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fiske; Robert	Wayland	MA		

US-CL-CURRENT: 717/171; 713/2, 717/174

ABSTRACT:

A method and processor program product for performing an upgrade of a program on a processor are provided. An upgraded version of the program is received into the processor and a backup of the program is created in memory associated with the processor. The upgraded version of the program is then installed and the processor is rebooted. Prior to initializing other application drivers, a reboot driver checks if the state of the processor is correct. If the reboot driver determines that the state of the processor is incorrect, the system reverts to the backup of the program. In alternative embodiments, if the reboot driver determines that the state is incorrect, the system inhibits the loading of other application drivers. Additionally, the reboot driver may set a counter to indicate the number of times that the processor is rebooted. If the counter has not reached a predetermined threshold, the system attempts to reboot the processor with the upgraded version of the program. If the counter reaches the predetermined threshold, the system reverts to the backup of the program.

30 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Detailed Description](#) | [Claims](#) | [KMM](#) | [Drawn By](#)

27. Document ID: US 6301710 B1

L5: Entry 27 of 51

File: USPT

Oct 9, 2001

US-PAT-NO: 6301710

DOCUMENT-IDENTIFIER: US 6301710 B1

TITLE: System and method for creating a substitute registry when automatically installing an update program

DATE-ISSUED: October 9, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fujiwara; Nobuyuki	Kanagawa			JP

US-CL-CURRENT: 717/175; 707/10, 707/203, 709/203, 709/221, 717/176, 717/178

ABSTRACT:

A system and method for creating a substitute registry when automatically installing an update program preferably comprises a plurality of computer systems

that are connected to a distributed network such as the Internet. Download modules on the computer systems automatically access and download update programs from the distributed network. Install modules then automatically create unique substitute registries that correspond to the downloaded update programs. Finally, the install module loads the update programs onto the computer systems to complete the software installation procedure, in accordance with the present invention.

38 Claims, 11 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 11

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Document](#) | [Claims](#) | [KWMC](#) | [Drawn D](#)

28. Document ID: US 6289511 B1

L5: Entry 28 of 51

File: USPT

Sep 11, 2001

US-PAT-NO: 6289511

DOCUMENT-IDENTIFIER: US 6289511 B1

**** See image for Certificate of Correction ****

TITLE: Method and system for distributing software in a telecommunications network

DATE-ISSUED: September 11, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hubinette; Ulf	Linkoping			SE

US-CL-CURRENT: 717/173; 709/223

ABSTRACT:

A method and system of a computer network enables efficient and reliable software distribution by disbursing the responsibility for loading software. A computer network is composed of, for example, a central managing station (CMS), multiple major network elements (NEs), and possibly multiple subordinated NEs (S-NEs) connected to major NEs. The CMS and the NEs are preferably connected to a network, such as an X.25 network. The CMS includes a load manager (LM) and at least one software unit to be distributed to multiple NEs and/or S-NEs. Multiple NEs include a load agent (LA), which aids the LM with software distribution. At least one particular LA receives instructions as well as the software unit from the CMS under control of the LM and via the network. The particular LA is then responsible for loading the software unit onto other NEs, either fully or partially. The process of upgrading and/or installing software is therefore delegated and disbursed between and among multiple NEs.

30 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Document](#) | [Claims](#) | [KWMC](#) | [Drawn D](#)

29. Document ID: US 6289510 B1

L5: Entry 29 of 51

File: USPT

Sep 11, 2001

US-PAT-NO: 6289510

DOCUMENT-IDENTIFIER: US 6289510 B1

TITLE: Online program-updating system and computer-readable recording medium storing a program-updating program

DATE-ISSUED: September 11, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nakajima; Ryoetsu	Kawasaki			JP

US-CL-CURRENT: 717/170

ABSTRACT:

There is provided an online program-updating system which is capable of updating contents of a program without interrupting any services provided by the system. A management information storage section stores management information including operative status flags each indicative of the operative status of a corresponding program module and information of a version number associated with the corresponding program module. A program execution section updates operative status flags based on the operative statuses of the program modules. A version number check section makes a comparison between the version number of each program module of a revised program and that of each program module of the existing program in response to a download request, and determines that an update of the program is required, if the version number of the program module of the revised program is newer. An operative status determination section checks on the operative status flag associated with each program module in the existing program, and determines that the update of the program is permitted, if the program module is an "inactive" operative status. A program transfer section executes download of a program module the update of which is determined to be required and at the same time permitted.

10 Claims, 15 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 15

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KANC](#) | [Drawn D](#)

30. Document ID: US 6282709 B1

L5: Entry 30 of 51

File: USPT

Aug 28, 2001

US-PAT-NO: 6282709

DOCUMENT-IDENTIFIER: US 6282709 B1

TITLE: Software update manager

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Reha; Mark Keith	San Jose	CA		
Morris; Charles F.	San Jose	CA		

US-CL-CURRENT: 717/175; 717/178

ABSTRACT:

A method and apparatus for checking/updating existing software on a user's computer utilizes a graphical user interface (GUI). The GUI enables the user, without knowing what software exists on the computer, to download a text file from a remote server and check whether the software on the remote server is contained on the user's computer. The user can also download and automatically install a new or updated program via the GUI.

25 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Print](#) | [Claims](#) | [KWMC](#) | [Drawn D](#)

31. Document ID: US 6256773 B1

L5: Entry 31 of 51

File: USPT

Jul 3, 2001

US-PAT-NO: 6256773

DOCUMENT-IDENTIFIER: US 6256773 B1

TITLE: System, method and article of manufacture for configuration management in a development architecture framework

DATE-ISSUED: July 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/121; 707/203, 717/168

ABSTRACT:

A system, method, and article of manufacture are provided for affording consistency in a development architecture framework as components in the framework change. A reference program code is provided and a plurality of sets of updated program code are received which represent different versions of the program code. The sets of the updated program code are compared with the reference program code in order to identify information relating to changes and the information is classified in relation to the changes. Tools are also provided for managing the different versions of the program code.

20 Claims, 14 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 14

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Print](#) | [Claims](#) | [KMC](#) | [Drawn D.](#)

32. Document ID: US 6237144 B1

L5: Entry 32 of 51

File: USPT

May 22, 2001

US-PAT-NO: 6237144

DOCUMENT-IDENTIFIER: US 6237144 B1

TITLE: Use of relational databases for software installation

DATE-ISSUED: May 22, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Delo, John C.	Bellevue	WA		

US-CL-CURRENT: 717/174; 707/10, 707/103Z, 707/3, 709/221, 717/106, 717/115

ABSTRACT:

A method and system for installing computer programs is provided where installation is accomplished based on an "as complete" description of the installed features, components and resources of the computer program. The necessary files and components required for installation of a given feature or component are determined by marking for installation any components which are not presently installed, preparing a script of required installation executions, and then executing the instructions to install the necessary files or components. Components are marked for installation or un-installation in temporary columns and rows which are dynamically added to data tables used to identify components and features which are available for installation. Individual components of a feature may be added or removed by simply marking that component for installation or removal. When the instructions in the installation script are executed, that particular component will be installed or removed according to the instructions. Installation of a given software application is streamlined because any component of a program application to be installed which is already installed on the user's computer need not be reinstalled upon the installation of the desired feature. Only components or files thereof which must be installed in addition to previously installed components or files need be installed for the installation of the software application program.

24 Claims, 6 Drawing figures

Exemplary Claim Number: 23

Number of Drawing Sheets: 5

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Print](#) | [Claims](#) | [KMC](#) | [Drawn D.](#)

33. Document ID: US 6182285 B1

US-PAT-NO: 6182285
DOCUMENT-IDENTIFIER: US 6182285 B1

TITLE: Method and apparatus for generating a default list

DATE-ISSUED: January 30, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bleizeffer; Terry M.	Santa Cruz	CA		
Church; Nathan D.	Seattle	WA		
Devine; Kathryn W.	Morgan Hill	CA		
Hughes, Jr.; Virginia W.	Hollister	CA		
Kilburn; Barbara J.	Saratoga	CA		
Shough; David E.	San Jose	CA		

US-CL-CURRENT: 717/170; 714/42

ABSTRACT:

A method and apparatus for compensating for deficiencies existing in programs to assist a user through installing a program. Polling the status of jobs requested by the user of a workstation is done so that the user may eventually be provided with status reports regarding the jobs being executed. The user can set parameters during loading of System Modification Program Extended (SMPE) libraries, install, migrate, fallback, remigrate and update procedures for the program. An indication is provided to a user of a workstation as steps of a task have been completed by the user. The health of catalog and directory databases may be verified before a migrate procedure is performed. The user of the program can be informed regarding parameters whose default values have changed, which parameters are of particular concern to the specific user.

24 Claims, 21 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 21

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KWMC](#) | [Drawn D.](#)

34. Document ID: US 6113652 A

US-PAT-NO: 6113652
DOCUMENT-IDENTIFIER: US 6113652 A

TITLE: Communications network equipment capable of non-disruptive software upgrade

DATE-ISSUED: September 5, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lysik; John L.	Burlington	CT		
Danenberg; L. David	Woodbury	CT		
Chagnon; James C.	Southington	CT		

US-CL-CURRENT: 717/170; 709/223

ABSTRACT:

In accord with the objects of the invention, methods and apparatus for controlling communications network equipment are provided. The apparatus is used in a telecommunications network having a plurality of coupled nodes and a network controller coupled to at least one of those nodes, and generally comprises a node apparatus having a backplane and a plurality of functional cards coupled to the backplane. Each of the functional cards has a processor and memory for storing software which is used by the processor. One of the functional cards has memory for storing a current running version of software for each of the other of the plurality of functional cards, and a receiver and memory for storing in background incoming updated versions of software for the plurality of functional cards. The updated versions of software are provided in the overhead portion of the telecommunications signal frame, thereby being non-disruptive to the system. Upon command of the network controller, the updated versions of software are provided in foreground by the functional card having the receiver to the respective software memories of the other cards, thereby causing the other cards to run the updated versions of the software. The updating occurs simultaneously in all nodes.

31 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KWMC](#) | [Drawn D.](#)

35. Document ID: US 6074434 A

L5: Entry 35 of 51

File: USPT

Jun 13, 2000

US-PAT-NO: 6074434

DOCUMENT-IDENTIFIER: US 6074434 A

TITLE: Selection of code updates, data updates or new data for client

DATE-ISSUED: June 13, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Cole; Gary Lee	Endicott	NY		
Engleman; Scott Phillip	Vestal	NY		
Pritko; Steven Michael	Endicott	NY		

US-CL-CURRENT: 717/173; 717/170

ABSTRACT:

A server computer selects code updates to download to a client computer as follows. The server computer identifies code updates which are consistent with basic system characteristics of the client computer. Then, the server computer sends to the client computer one or more "recognizer" programs which execute in the client computer to determine whether the client computer has a version other than a current version of the consistent code updates. The client sends the results to the server computer which generates a list of those code updates which are consistent with the basic system characteristics, and represent programs that exist on the client computer for which an update would be appropriate. The server computer also identifies new data which is consistent with attributes of a user of the client computer. Then, the server computer sends to the client computer one or more recognizer programs which execute in the client computer to determine whether the client computer already has the consistent new data. The client sends the results to the server computer which generates a list of that new data which is consistent with the user attributes and not currently resident in the client computer.

20 Claims, 12 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 12

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Assignee](#) | [Attorneys](#) | [Claims](#) | [KMC](#) | [Drawn Da](#)

36. Document ID: US 6067582 A

L5: Entry 36 of 51

File: USPT

May 23, 2000

US-PAT-NO: 6067582

DOCUMENT-IDENTIFIER: US 6067582 A

TITLE: System for installing information related to a software application to a remote computer over a network

DATE-ISSUED: May 23, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Smith; Benjamin Hewitt	Belmont	MA		
Smith; Fred Hewitt	Belmont	MA		

US-CL-CURRENT: 710/5; 709/203, 717/177

ABSTRACT:

A system and method is disclosed for distributing, registering and purchasing software application and other digital information over a network. Each software application is embedded with an agent module which communicates with a remote server module in a server attached to the network. The server module interacts with the user that is requesting installation of the software application and upon verification of billing or other constraints, the server module enables the agent module to proceed with installation. Subsequent to installation, the agent module monitors the server module and informs the user if an update to the software application is available.

24 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

[Full | Title | Citation | Front | Review | Classification | Date | Reference |  |  | Claims | KMC | Drawn Ds]

37. Document ID: US 6064821 A

L5: Entry 37 of 51

File: USPT

May 16, 2000

US-PAT-NO: 6064821

DOCUMENT-IDENTIFIER: US 6064821 A

TITLE: Method and apparatus for polling job status on a mainframe system

DATE-ISSUED: May 16, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Shough; David E.	San Jose	CA		
Church; Nathan D.	Seattle	WA		

US-CL-CURRENT: 717/174; 709/203, 709/220, 715/700, 715/866, 715/961

ABSTRACT:

A method and apparatus for compensating for deficiencies existing in programs to assist a user through installing a program. Polling the status of jobs requested by the user of a workstation is done so that the user may eventually be provided with status reports regarding the jobs being executed. The user can set parameters during loading of SMPE libraries, install, migrate, fallback, remigrate and update procedures for the program. An indication is provided to a user of a workstation as steps of a task have been completed by the user. The health of catalog and directory databases may be verified before a migrate procedure is performed. The user of the program can be informed regarding parameters whose default values have changed, which parameters are of particular concern to the specific user.

21 Claims, 21 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 21

[Full | Title | Citation | Front | Review | Classification | Date | Reference |  |  | Claims | KMC | Drawn Ds]

38. Document ID: US 6052531 A

L5: Entry 38 of 51

File: USPT

Apr 18, 2000

US-PAT-NO: 6052531

DOCUMENT-IDENTIFIER: US 6052531 A

TITLE: Multi-tiered incremental software updating

DATE-ISSUED: April 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Waldin, Jr.; Ray Soon	Los Angeles	CA		
Nachenberg; Carey	Northridge	CA		

US-CL-CURRENT: 717/170; 707/10, 707/203, 709/203, 709/219

ABSTRACT:

A software application (110) is updated to a newer version by means of incremental update patches (122). The incremental update patches (122) each contain that information necessary to transform one version of an application to another version. Any version of an application (110) may be upgraded to any other version of the application, through the use of a series of incremental update patches (122). The appropriate incremental update patches (122) are distributed in a multi-tiered manner, such that some update patches (122) update the application (110) by only one version, and others update the application (110) by several versions.

22 Claims, 9 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KMNC	Drawn D.
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	----------

39. Document ID: US 5999740 A

L5: Entry 39 of 51

File: USPT

Dec 7, 1999

US-PAT-NO: 5999740

DOCUMENT-IDENTIFIER: US 5999740 A

TITLE: Updating mechanism for software

DATE-ISSUED: December 7, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rowley; David John	Warrington			GB

US-CL-CURRENT: 717/173; 709/221, 709/223

ABSTRACT:

A computer has a memory storing a number of software applications, and a registration file, indicating which application versions are currently installed in the memory. A software update mechanism in the computer accesses a remote file server to obtain a release file containing a list of software applications available from the remote server, and compares the release file with the registration file to determine which of the installed applications have upgrades available. When a user selects an application for upgrading, and the update mechanism accesses the remote file server to obtain a manifest file containing

details of the application files required to form an updated version of this applications. The manifest file is used to determine which of the required application files are already available in the computer; and only those application files that are not already available in the computer are accessed and installed in the memory.

20 Claims, 10 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KMC](#) | [Drawn D](#)

40. Document ID: US 5909581 A

L5: Entry 40 of 51

File: USPT

Jun 1, 1999

US-PAT-NO: 5909581

DOCUMENT-IDENTIFIER: US 5909581 A

TITLE: Automatic software updating method

DATE-ISSUED: June 1, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Park; Seong-Kab	Kuri			KR

US-CL-CURRENT: 717/170

ABSTRACT:

An automatic software updating method in a communication network including a host computer, a plurality of branch processing computers each connected to the host computer for serving as servers, and a plurality of personal computers connected to a corresponding branch processing computer is provided. The method includes the steps of: (a) uploading a new software together with a version-up table including a version code to the host computer; (b) downloading the version-up table uploaded in step (a) to the branch processing computer; (c) determining whether to download the software using the version code of the version-up table downloaded in step (b); (d) downloading the software according to the result of step (c); and (e) installing the software downloaded in step (d) in the personal computer. Therefore, an out-of-date software of computer systems dispersedly arranged in the network can be automatically updated by a partially corrected or newly developed software.

24 Claims, 3 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KMC](#) | [Drawn D](#)

41. Document ID: US 5764992 A

US-PAT-NO: 5764992

DOCUMENT-IDENTIFIER: US 5764992 A

TITLE: Method and apparatus for automatic software replacement

DATE-ISSUED: June 9, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kullick; Steven	Saratoga	CA		
Titus; Diane	Santa Clara	CA		

US-CL-CURRENT: 717/170; 717/171

ABSTRACT:

A software program running on a computer automatically replaces itself with a newer version in a completely automated fashion, without interruption of its primary function, and in a manner that is completely transparent to the user of the computer. This is achieved by means of a logic module that is incorporated into programs. The logic module performs the functions of locating and identifying other versions of its associated program, determining whether the other versions are older or newer than the currently running version, and replacing older versions of itself with a newer version. As part of this operation, the logic module can copy the newer version to its current location, move the older version to a secondary location, and remove older versions of itself that have been replaced by a newer version. The new version that is to replace an older version can reside on an individual computer, or can be present on a server to which a number of computers are connected via a network. With this arrangement, software upgrades can be effected in an efficient and automatic manner, without resort to any external resources.

30 Claims, 12 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Dra
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	-----

 42. Document ID: US 5754785 A

US-PAT-NO: 5754785

DOCUMENT-IDENTIFIER: US 5754785 A

TITLE: Communications network equipment

DATE-ISSUED: May 19, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lysik; John L.	Burlington	CT		
Danenberg; L. David	Woodbury	CT		
Chagnon; James C.	Southington	CT		

US-CL-CURRENT: 709/222; 717/170, 717/173

ABSTRACT:

In accord with the objects of the invention, methods and apparatus for controlling communications network equipment are provided. The apparatus is used in a telecommunications network having a plurality of coupled nodes and a network controller coupled to at least one of those nodes, and generally comprises a node apparatus having a backplane and a plurality of functional cards coupled to the backplane. Each of the functional cards has a processor and memory for storing software which is used by the processor. One of the functional cards has memory for storing a current running version of software for each of the other of the plurality of functional cards, and a receiver and memory for storing in background incoming updated versions of software for the plurality of functional cards. The updated versions of software are provided in the overhead portion of the telecommunications signal frame, thereby being non-disruptive to the system. Upon command of the network controller, the updated versions of software are provided in foreground by the functional card having the receiver to the respective software memories of the other cards, thereby causing the other cards to run the updated versions of the software. The updating occurs simultaneously in all nodes.

1 Claims, 2 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 2

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KJC](#) | [Drawn D](#)

43. Document ID: US 5752042 A

L5: Entry 43 of 51

File: USPT

May 12, 1998

US-PAT-NO: 5752042

DOCUMENT-IDENTIFIER: US 5752042 A

TITLE: Server computer for selecting program updates for a client computer based on results of recognizer program(s) furnished to the client computer

DATE-ISSUED: May 12, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Cole; Gary Lee	Endicott	NY		
Engleman; Scott Phillip	Vestal	NY		
Pritko; Steven Michael	Endicott	NY		

US-CL-CURRENT: 717/173; 707/203, 709/219, 709/221

ABSTRACT:

A server computer selects code updates to download to a client computer. The server computer identifies code updates which are consistent with basic system characteristics of the client computer. Then, the server computer sends to the client computer one or more "recognizer" programs or one or more addresses outside of the client computer of the one or more recognizer programs which execute in the client computer to determine whether the client computer has a version other than a current version of the identified code updates. The client sends the results to the server computer which generates a list of code updates which are consistent with the basic system characteristics, representing programs that exist on the client computer for which an update would be appropriate. Next, the server computer sends the list or information about the list to the client computer. A user at the client computer selects from the list and sends the selections to the server computer. In response, the server computer sends addresses of the selected code updates to the client computer and the client computer downloads the selected code updates from a content server. At some later time, the recognizer programs corresponding to the selected code updates are executed again before the selected code updates are installed in the client computer.

30 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search Patent](#) | [Search Docket](#) | [Claims](#) | [DOCID](#) | [Drawn D.](#)

44. Document ID: US 5732275 A

L5: Entry 44 of 51

File: USPT

Mar 24, 1998

US-PAT-NO: 5732275

DOCUMENT-IDENTIFIER: US 5732275 A

TITLE: Method and apparatus for managing and automatically updating software programs

DATE-ISSUED: March 24, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kullick; Steven	Saratoga	CA		
Titus; Diane	Santa Clara	CA		

US-CL-CURRENT: 717/170; 713/1, 717/173

ABSTRACT:

A software program running on a computer is automatically managed, monitored and updated with a newer version in a completely automated fashion, without interruption of its primary function, and in a manner that is completely transparent to the user of the computer. This is achieved by means of a control module that performs the functions of locating and identifying other versions of its associated program, determining whether the other versions are older or newer than currently stored versions, and downloading a newer version. Multiple versions

of the program can remain accessible on the computer, and the control module manages the launching of a particular version that may be required. Statistical data relating to the launching and operation of the program is collected, and uploaded to a central location on a regular basis.

25 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Specification	Claims	TOC	Drawing
------	-------	----------	-------	--------	----------------	------	-----------	----------	---------------	--------	-----	---------

45. Document ID: US 5701492 A

L5: Entry 45 of 51

File: USPT

Dec 23, 1997

US-PAT-NO: 5701492

DOCUMENT-IDENTIFIER: US 5701492 A

TITLE: Fail-safe flashing of EPROM

DATE-ISSUED: December 23, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wadsworth; Robert D.	Costa Mesa	CA		
Danknick; Daniel A.	Orange	CA		

US-CL-CURRENT: 717/173; 711/103, 711/152, 711/156, 713/2

ABSTRACT:

A fail-safe flash for an EPROM that is arranged into separately erasable sectors proceeds in a stepwise procedure. The EPROM is divided into at least three regions including a boot block region, a file region and a directory region. The boot block region stores at least two independently erasable boot blocks, only one of which is designated as a current boot block, which performs scanning of the directory region so as to determine a current directory in the directory region. The file region stores all executables, including executables designated as critical and non-critical. The directory region stores multiple directories, only one of which is a current directory, which designates which is the current boot block, which of the files stored in the file region are available for loading for execution, as well as which of the files are critical and which are non-critical. In the first step to flash the EPROM, the unused region of the boot block is updated with the new boot block. In the second step of flashing the EPROM, critical files are flashed over non-critical files, and after correct installation of the newly-flashed critical files is confirmed, a new directory is written into the directory region referring only to the newly-flashed critical files. In the third step of flashing the EPROM, non-critical files are flashed, whereafter a new directory is written to the directory region referring to the newly-flashed critical files as well as to the newly-flashed non-critical files.

14 Claims, 8 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

46. Document ID: US 5649112 A

L5: Entry 46 of 51

File: USPT

Jul 15, 1997

US-PAT-NO: 5649112

DOCUMENT-IDENTIFIER: US 5649112 A

TITLE: Method and apparatus for modifying microcode in a distributed nodal network while the network continues operation

DATE-ISSUED: July 15, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yeager; John D.	San Jose	CA		
Ho; Lawrence Y.	Morgan Hill	CA		
Stevens; Chester R.	San Jose	CA		
Brady; James T.	San Jose	CA		
Wang; David T.	San Jose	CA		

US-CL-CURRENT: 709/220; 709/212, 717/170

ABSTRACT:

Updating of control code is accomplished in multiple nodes of a computing system while the computing system remains in operation. Each node includes a processor, memory, a first version of a control code unit and an engineering change level indication for the control code unit. The method comprises the steps of: installing a revised version of the control code unit with converter code modules in a first node, the converter code modules enabling and performing first and second interface functions during communications between the first node and other nodes in the system. The first node is then operated to perform a function which requires communication with other nodes, the converter code module in the first node initially determining an engineering change level value stored in another node and, if the engineering change level values in the nodes match, communicating with the other node through the first interface function. If the engineering change level values are found not to match, communications occur with the other node through the use of the second interface function, enabling both nodes to communicate even though different level code changes are present. A sequencer is also provided in the computing system which enables updating of all nodes in a specified sequence so as to simplify the requirements placed upon the converter code module.

9 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

47. Document ID: US 5642417 A

L5: Entry 47 of 51

File: USPT

Jun 24, 1997

US-PAT-NO: 5642417

DOCUMENT-IDENTIFIER: US 5642417 A

TITLE: Virtualized installation of material

DATE-ISSUED: June 24, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Stringer; John W.	Santa Cruz	CA		

US-CL-CURRENT: 717/174; 700/245, 703/6, 713/100

ABSTRACT:

Virtualization converts a normally unidirectional process into a bi-directional process to permit automatic install and uninstall of new material, addition and/or modification of target materials while preserving the state of target material, and conversion of materials having a specific media characteristic into an extended range of media characteristics. Applications include temporary retooling of assembly line for short production runs, associating the installation of materials with the mounting of removable media and the uninstalling materials with the unmounting of removable media, and repurposing stand-alone applications for client-server environments without requiring modifications to the design of the stand-alone application.

21 Claims, 8 Drawing figures

Exemplary Claim Number: 7

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Search](#) | [Get PDF](#) | [Claims](#) | [RWC](#) | [Drawn D](#)

48. Document ID: US 5630135 A

L5: Entry 48 of 51

File: USPT

May 13, 1997

US-PAT-NO: 5630135

DOCUMENT-IDENTIFIER: US 5630135 A

TITLE: Multiple-execution method of multiple-version programs and computer system therefor

DATE-ISSUED: May 13, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Orimo; Masayuki	Kawasaki			JP
Mori; Kinji	Machida			JP

Kawano; Katsumi	Kawasaki	JP
Hirasawa; Shigeki	Sagamihara	JP
Fujise; Hiroshi	Yokohama	JP
Suzuki; Hitoshi	Owariasahi	JP
Nakamura; Tomoaki	Katsuta	JP

US-CL-CURRENT: 718/106; 717/170, 719/313

ABSTRACT:

A method for executing programs in a distributed processing system includes executing in a plurality of first processors different programs which perform the same function. Messages, containing data representing execution result output from said different programs and attribute information indicating function to be performed by the different programs, are transmitted from the first processors to a network. A second processor then receives these messages and selects a message therefrom based on an analysis of the attribute information contained in all of the received messages. The second processor then executes an application program using the data contained in the selected message. Alternatively, the message-selecting step is made based on both the attribute information and data contained in the messages received in the second processor. Each of the messages from the first processors may include graphic image data which can be analyzed and shown on a display.

11 Claims, 10 Drawing figures

Exemplary Claim Number: 8

Number of Drawing Sheets: 5

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstracts](#) | [Detailed Abstracts](#) | [Claims](#) | [KOMC](#) | [Drawn D.](#)

49. Document ID: US 5210854 A

L5: Entry 49 of 51

File: USPT

May 11, 1993

US-PAT-NO: 5210854

DOCUMENT-IDENTIFIER: US 5210854 A

TITLE: System for updating program stored in EEPROM by storing new version into new location and updating second transfer vector to contain starting address of new version

DATE-ISSUED: May 11, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Beaverton; Arthur J.	Maynard	MA		
Hunt; Thomas E.	Brookline	NH		

US-CL-CURRENT: 717/174; 711/103, 711/152, 711/173, 717/168

ABSTRACT:

Firmware resident in electrically erasable programmable read only memory ("EEPROM") can be updated by a user while maintaining the intelligence of a computer system during the updating process by a control logic device. The control logic device decodes address and control signals to provide a hardware partitioning of the firmware resident in the EEPROMs to prevent writing to protected partitions of the firmware. Transfer vectors are used to provide indirect accessing of subroutines resident in the firmware. During an updating process, a new version of a subroutine is stored in a free area in the EEPROMs before the transfer vector pointing to the old version of the subroutine is updated. The window of vulnerability to errors during the updating process is minimized by only updating a page of memory containing the transfer vector that points to the old version of the subroutine after the new version has been stored.

7 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#)  [Claims](#) | [KWMC](#) | [Drawn De](#)

50. Document ID: US 5008814 A

L5: Entry 50 of 51

File: USPT

Apr 16, 1991

US-PAT-NO: 5008814

DOCUMENT-IDENTIFIER: US 5008814 A

TITLE: Method and apparatus for updating system software for a plurality of data processing units in a communication network

DATE-ISSUED: April 16, 1991

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mathur; Ashish	Sunnyvale	CA		

US-CL-CURRENT: 709/221; 709/219, 717/170, 717/171, 717/172, 717/174

ABSTRACT:

In maintaining a communication network of processing units distributed in multiple nodes linked by communication channels, system software in a plurality of data processing units is updated by first installing the updated software in a first node. The updated software is transmitted through the network to other nodes. A trial use of the updated software is initiated in the nodes. If failures of the updated software are detected in a node, that node will be restored to the original software version. If the trial use of the updated software is completed successfully in a node, the updated version will be installed as a preferred operational version in the node. a

21 Claims, 8 Drawing figures

Exemplary Claim Number: 12

Number of Drawing Sheets: 8

Terms

Documents

L4 and install

51

Display Format:

[Previous Page](#)

[Next Page](#)

[Go to Doc#](#)